

Отчёт по лабораторной работе 15

Настройка сетевого журналирования

Авдадаев Джамал Геланиевич

Содержание

1	Введение	5
1.1	Цель работы	5
2	Процесс работы	6
2.1	Настройка сервера сетевого журнала	6
2.1.1	Создание конфигурационного файла	6
2.1.2	Перезапуск rsyslog и проверка открытых портов	6
2.2	Настройка клиента сетевого журнала	7
2.2.1	Создание файла конфигурации	7
2.3	Просмотр логов	8
2.3.1	Просмотр системного журнала на сервере	8
2.3.2	Просмотр журналов через графическую утилиту	8
2.3.3	Установка консольного просмотрщика lnav	9
2.4	Подготовка окружения Vagrant для автоконфигурации	10
2.4.1	Сервер: экспорт конфигураций и создание скрипта	10
2.4.2	Клиент: экспорт конфигураций и создание скрипта	10
3	Итоги	12
3.1	Вывод	12
3.2	Контрольные вопросы	12

Список иллюстраций

2.1	Создание конфигурации netlog-server.conf	6
2.2	Порты rsyslog после запуска TCP-сервера	7
2.3	Конфигурация клиента netlog-client.conf	7
2.4	Пример входящих сообщений в /var/log/messages	8
2.5	Просмотр ресурсов и журнала в gnome-system-monitor	9
2.6	Ошибка установки lnav	9
2.7	Создание структуры netlog для сервера	10
2.8	Скрипт провизининга сервера	10
2.9	Экспорт конфигурации клиента	11
2.10	Скрипт провизининга клиента	11

Список таблиц

1 Введение

1.1 Цель работы

Получение навыков по работе с журналами системных событий.

2 Процесс работы

2.1 Настройка сервера сетевого журнала

2.1.1 Создание конфигурационного файла

Для включения сетевого приёма журналов по TCP-порту 514 на сервере был создан конфигурационный файл `netlog-server.conf`:

В файл внесены параметры загрузки модуля и запуска TCP-сервера:

```
$ModLoad imtcp
```

```
$InputTCPServerRun 514
```

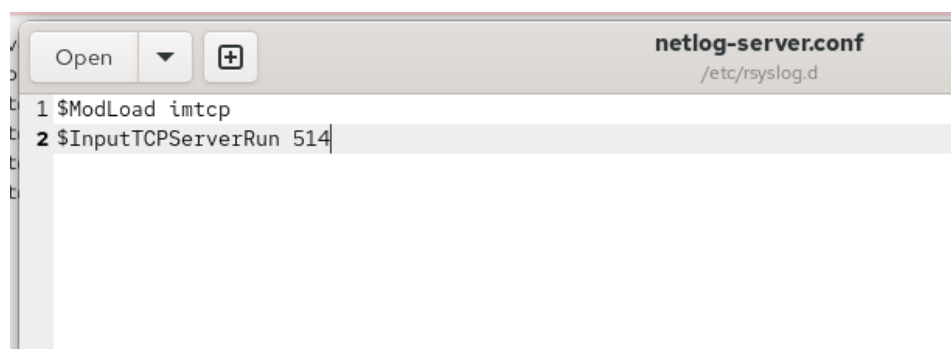


Рис. 2.1: Создание конфигурации `netlog-server.conf`

2.1.2 Перезапуск `rsyslog` и проверка открытых портов

После обновления конфигурации служба `rsyslog` была перезапущена, а список слушающих портов проверен с помощью утилиты `lsof`.

```

>client.dgavdadaev.net:43644 (ESTABLISHED)
rsyslogd 14826 root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14829 in:imjour root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14829 in:imjour root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14830 in:imtcp root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14830 in:imtcp root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14831 w0/imtcp root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14831 w0/imtcp root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14832 w1/imtcp root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14832 w1/imtcp root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14833 rs:main root 4u IPv4 59870 0t0 TCP *:shell (LISTEN)
rsyslogd 14826 14833 rs:main root 5u IPv6 59871 0t0 TCP *:shell (LISTEN)
[root@server.dgavdadaev.net rsyslog.d]# firewall-cmd --add-port=514/tcp
success
[root@server.dgavdadaev.net rsyslog.d]# firewall-cmd --add-port=514/tcp --permanent
success
[root@server.dgavdadaev.net rsyslog.d]# █

```

Рис. 2.2: Порты rsyslog после запуска TCP-сервера

Разрешение входящих TCP-соединений на порт 514.

2.2 Настройка клиента сетевого журнала

2.2.1 Создание файла конфигурации

На клиенте был создан файл `netlog-client.conf`.

В файл `netlog-client.conf` добавлена строка для отправки всех сообщений на сервер:

```
*.* @@server.dgavdadaev.net:514
```



Рис. 2.3: Конфигурация клиента netlog-client.conf

Служба rsyslog была перезапущена для применения изменений.

2.3 Просмотр логов

2.3.1 Просмотр системного журнала на сервере

Для наблюдения за входящими записями использовался просмотр файла `/var/log/messages`.

```
Dec 11 11:04:12 server systemd[1]: systemd-coredump@81-13750-0.service: Deactivated successfully.
Dec 11 11:04:16 client kernel: traps: VBoxClient[13749] trap int3 ip:41dd1b sp:7f3ac9977cd0 error:0 in VBoxClient[1dd1b,400000+bb000]
Dec 11 11:04:16 client systemd-coredump[13750]: Process 13746 (VBoxClient) of user 1001 terminated abnormally with signal 5/TRAP, process
ing...
Dec 11 11:04:16 client systemd[1]: Started systemd-coredump@81-13750-0.service - Process Core Dump (PID 13750/UID 0).
Dec 11 11:04:16 client systemd-coredump[13751]: Process 13746 (VBoxClient) of user 1001 dumped core.#012#012Module libXau.so.6 from rpm l
ibXau-1.0.11-8.el10.x86_64#012Module libxcb.so.1 from rpm libxcb-1.17.0-3.el10.x86_64#012Module libX11.so.6 from rpm libX11-1.8.10-1.el10
.x86_64#012Module libffi.so.8 from rpm libffi-3.4.4-9.el10.x86_64#012Module libwayland-client.so.0 from rpm wayland-1.23.0-2.el10.x86_64#
012Stack trace of thread 13749:#012#0 0x0000000041dd1b n/a (n/a + 0x0)#012#1 0x00000000041dc94 n/a (n/a + 0x0)#012#2 0x000000000045
041c n/a (n/a + 0x0)#012#3 0x0000000004355d0 n/a (n/a + 0x0)#012#4 0x00007f3ad801eb68 start_thread (libc.so.6 + 0x94b68)#012#5 0x0000
7f3ad808f6bc __clone3 (libc.so.6 + 0x1056bc)#012#012Stack trace of thread 13746:#012#0 0x00007f3ad808d4bd syscall (libc.so.6 + 0x1034bd)
#012#1 0x0000000004344e2 n/a (n/a + 0x0)#012#2 0x000000000450066 n/a (n/a + 0x0)#012#3 0x000000000405123 n/a (n/a + 0x0)#012#4 0x0
0007f3ad7fb430e __libc_start_call_main (libc.so.6 + 0x2a30e)#012#5 0x00007f3ad7fb43c9 __libc_start_main@@GLIBC_2.34 (libc.so.6 + 0x2a3c9
)#012#6 0x0000000004044aa n/a (n/a + 0x0)#012ELF object binary architecture: AMD x86-64
Dec 11 11:04:16 client systemd[1]: systemd-coredump@81-13750-0.service: Deactivated successfully.
Dec 11 11:04:17 server systemd[1]: serial-getty@ttyS0.service: Deactivated successfully.
Dec 11 11:04:17 server systemd[14337]: Created slice background.slice - User Background Tasks Slice.
Dec 11 11:04:17 server systemd[14337]: Starting systemd-tmpfiles-clean.service - Cleanup of Host's Temporary Files and Directories
```

Рис. 2.4: Пример входящих сообщений в `/var/log/messages`

2.3.2 Просмотр журналов через графическую утилиту

На сервере под пользователем `user` была запущена программа `gnome-system-monitor`.

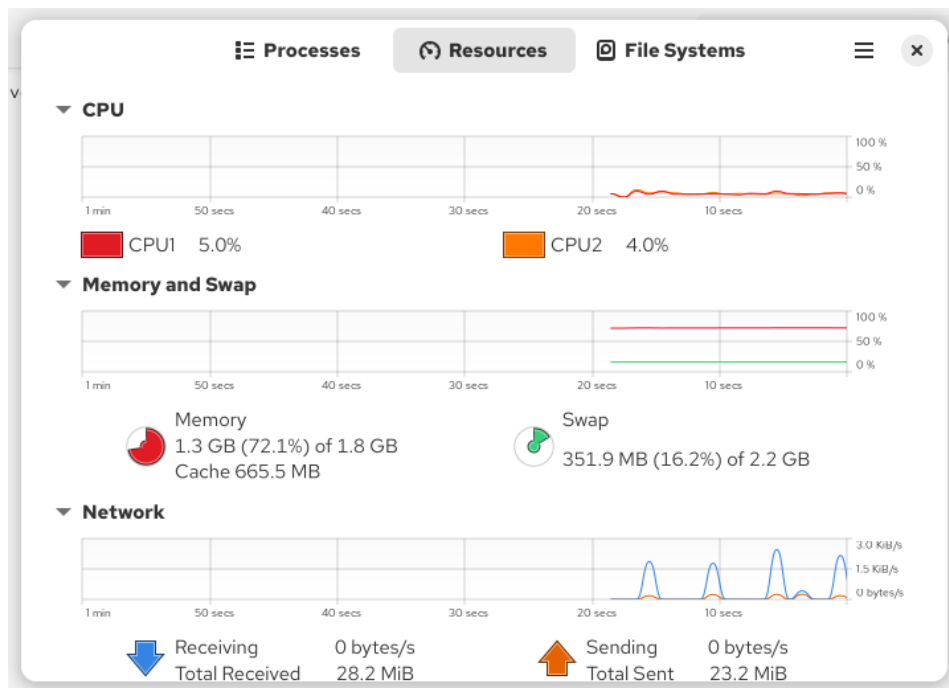


Рис. 2.5: Просмотр ресурсов и журнала в gnome-system-monitor

2.3.3 Установка консольного просмотрщика lnav

Попытка установки пакета lnav из стандартных репозиторийев завершилась ошибкой: пакет не найден.

```
[root@server.dgavdadaev.net ~]# dnf -y install lnav
Extra Packages for Enterprise Linux 10 - x86_64
Extra Packages for Enterprise Linux 10 - x86_64
Rocky Linux 10 - BaseOS
Rocky Linux 10 - BaseOS
Rocky Linux 10 - AppStream
Rocky Linux 10 - AppStream
Rocky Linux 10 - CRB
Rocky Linux 10 - CRB
Rocky Linux 10 - Extras
Rocky Linux 10 - Extras
No match for argument: 'lnav'
Error: Unable to find a match: lnav
[root@server.dgavdadaev.net ~]#
```

21 kB/s	34 kB	00:01
12 MB/s	5.6 MB	00:00
11 kB/s	4.3 kB	00:00
11 MB/s	4.1 MB	00:00
12 kB/s	4.3 kB	00:00
2.0 MB/s	2.0 MB	00:01
14 kB/s	4.3 kB	00:00
952 kB/s	484 kB	00:00
7.4 kB/s	3.1 kB	00:00
12 kB/s	4.8 kB	00:00

Рис. 2.6: Ошибка установки lnav

2.4 Подготовка окружения Vagrant для автоконфигурации

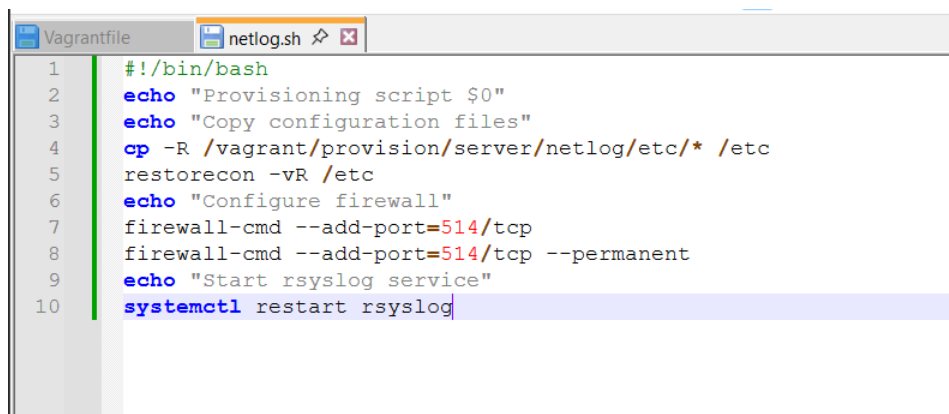
2.4.1 Сервер: экспорт конфигураций и создание скрипта

На сервере была создана директория `/vagrant/provision/server/netlog/etc/rsyslog.d`, в которую помещён файл `netlog-server.conf`.

```
[root@server.dgavdadaev.net rsyslog.d]#  
[root@server.dgavdadaev.net rsyslog.d]# cd /vagrant/provision/server/  
[root@server.dgavdadaev.net server]# mkdir -p /vagrant/provision/server/netlog/etc/rsyslog.d  
[root@server.dgavdadaev.net server]# cp -R /etc/rsyslog.d/netlog-server.conf /vagrant/provision/server/netlog/etc/rsyslog.d/  
[root@server.dgavdadaev.net server]# touch netlog.sh  
[root@server.dgavdadaev.net server]#
```

Рис. 2.7: Создание структуры netlog для сервера

Затем создан исполняемый файл `netlog.sh`, содержащий автоматизацию копирования конфигураций, настройки firewall и перезапуска rsyslog.



```
Vagrantfile netlog.sh  
1  #!/bin/bash  
2  echo "Provisioning script $0"  
3  echo "Copy configuration files"  
4  cp -R /vagrant/provision/server/netlog/etc/* /etc  
5  restorecon -vR /etc  
6  echo "Configure firewall"  
7  firewall-cmd --add-port=514/tcp  
8  firewall-cmd --add-port=514/tcp --permanent  
9  echo "Start rsyslog service"  
10 systemctl restart rsyslog
```

Рис. 2.8: Скрипт провижининга сервера

2.4.2 Клиент: экспорт конфигураций и создание скрипта

На клиенте была создана структура `/vagrant/provision/client/netlog/etc/rsyslog.d`, куда помещён файл `netlog-client.conf`.

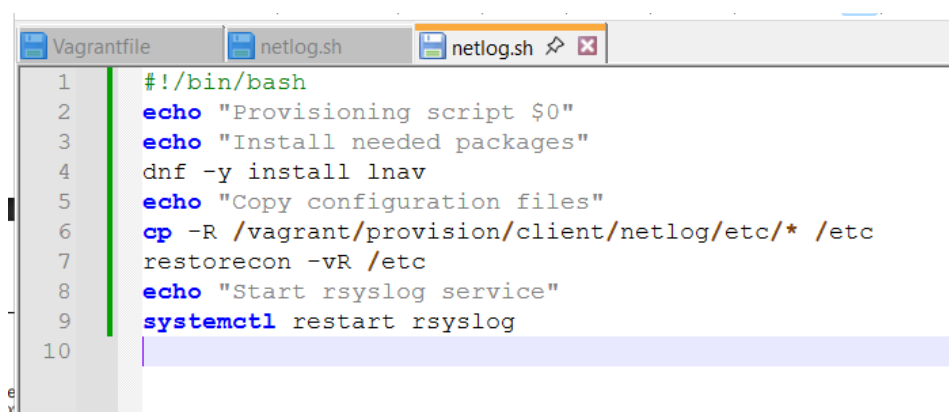
```

[root@client.dgavdadaev.net rsyslog.d]#
[root@client.dgavdadaev.net rsyslog.d]# cd /vagrant/provision/client/netlog/etc/rsyslog.d
-bash: cd: /vagrant/provision/client/netlog/etc/rsyslog.d: No such file or directory
[root@client.dgavdadaev.net rsyslog.d]# mkdir -p /vagrant/provision/client/netlog/etc/rsyslog.d
[root@client.dgavdadaev.net rsyslog.d]# cd /vagrant/provision/client/
[root@client.dgavdadaev.net client]# cp -R /etc/rsyslog.d/netlog-client.conf /vagrant/provision/c
netlog/etc/rsyslog.d/
[root@client.dgavdadaev.net client]# touch netlog.sh
[root@client.dgavdadaev.net client]# █

```

Рис. 2.9: Экспорт конфигурации клиента

Создан скрипт `netlog.sh`, выполняющий установку дополнительных утилит, копирование конфигураций и перезапуск `rsyslog`.



```

Vagrantfile  netlog.sh  netlog.sh
1  #!/bin/bash
2  echo "Provisioning script $0"
3  echo "Install needed packages"
4  dnf -y install lnav
5  echo "Copy configuration files"
6  cp -R /vagrant/provision/client/netlog/etc/* /etc
7  restorecon -vR /etc
8  echo "Start rsyslog service"
9  systemctl restart rsyslog
10

```

Рис. 2.10: Скрипт провижининга клиента

3 Итоги

3.1 Вывод

В ходе работы была выполнена настройка серверной и клиентской части rsyslog, обеспечена передача журналов по TCP-порту 514, а также подготовлены provisioning-скрипты для автоматизации конфигурации в среде Vagrant. Проверена корректная работа логирования, взаимодействие journald и rsyslog, а также механизмы просмотра журналов. Лабораторный стенд успешно функционирует и соответствует требованиям задания.

3.2 Контрольные вопросы

1. Какой модуль rsyslog вы должны использовать для приёма сообщений от journald?

Используется модуль **imjournal**, обеспечивающий прямую интеграцию rsyslog с journald.

2. Как называется устаревший модуль, который можно использовать для включения приёма сообщений журнала в rsyslog?

Устаревший модуль — **imuxsock**, основанный на механизме сокетов /dev/log.

3. Чтобы убедиться, что устаревший метод приёма сообщений из journald в rsyslog не используется, какой дополнительный параметр следует использовать?

Следует установить параметр **IMJournalIgnorePreviousMessages="on"**, что исклю-

чает подачу в rsyslog старых сообщений, перенаправленных journald через сокет.

4. В каком конфигурационном файле содержатся настройки, которые позволяют вам настраивать работу журнала?

Основные параметры журнала определяются в файле `/etc/rsyslog.conf` и включаемых файлах в директории `/etc/rsyslog.d/`.

5. Каким параметром управляется пересылка сообщений из journald в rsyslog?

Пересылку определяет параметр `ForwardToSyslog=` в файле `journald.conf`.

6. Какой модуль rsyslog вы можете использовать для включения сообщений из файла журнала, не созданного rsyslog?

Для чтения произвольных файлов используется модуль `imfile`.

7. Какой модуль rsyslog вам нужно использовать для пересылки сообщений в базу данных MariaDB?

Для работы с СУБД MariaDB применяется модуль `ommysql`.

8. Какие две строки вам нужно включить в rsyslog.conf, чтобы позволить текущему журнальному серверу получать сообщения через TCP?

```
$ModLoad imtcp
```

```
$InputTCPServerRun 514
```

9. Как настроить локальный брандмауэр, чтобы разрешить приём сообщений журнала через порт TCP 514?

Необходимо открыть порт:

```
firewall-cmd --add-port=514/tcp
```

```
firewall-cmd --add-port=514/tcp --permanent
```

И затем перезагрузить правила:

```
firewall-cmd --reload
```